

## CLAIMS

What is claimed is:

1. An apparatus for parsing an input data stream, comprising:
  - a first storage component operable to store a history buffer for containing an unencoded version of at least one previously encoded string;
  - a first comparison component operable to compare a string from said input data stream with said unencoded version of said at least one previously encoded string;
  - a second storage component operable to store: an indicator that there exist at least two matches found by said first comparison component, and tokens corresponding to said at least two matches;
  - a summing component operable to sum potential token lengths to provide total potential token lengths;
  - a second comparison component operable to compare said total potential token lengths;
  - a selection component operable to select a match corresponding to a shortest total token length to represent said string from said input data stream; and
  - an emitting component for emitting tokens representing said match corresponding to a shortest total token length.
2. An apparatus as claimed in claim 1, wherein said apparatus comprises a data compression apparatus.
3. An apparatus as claimed in claim 2, wherein said apparatus comprises an adaptive dictionary

based data compression apparatus.

4. An apparatus as claimed in claim 3, wherein said apparatus comprises a Lempel-Ziv data compression apparatus.

5. An apparatus as claimed in claim 1, wherein said apparatus comprises a data encryption apparatus.

6. An apparatus as claimed in claim 1, wherein said apparatus comprises a message digest generation apparatus.

7. A method for parsing an input data stream, comprising:

storing a history buffer for containing an unencoded version of at least one previously encoded string;

comparing a string from said input data stream with said unencoded version of said at least one previously encoded string;

storing: an indicator that there exist at least two matches found by said first comparison component, and tokens corresponding to said at least two matches;

summing potential token lengths to provide total potential token lengths;

comparing said total potential token lengths;

selecting a match corresponding to a shortest total token length to represent said string from said input data stream; and

emitting tokens representing said match corresponding to a shortest total token length.

8. A method as claimed in claim 7, wherein said tokens comprise compressed data corresponding to said at least two matches.

9. A method as claimed in claim 8, wherein said compressed data comprises adaptive dictionary based compressed data.

10. A method as claimed in claim 7, wherein said tokens comprise encrypted data corresponding to said at least two matches.

11. A method as claimed in claim 7, wherein said tokens comprise message digest data corresponding to said at least two matches.

12. A computer program product, tangibly embodied in a computer-readable medium, comprising computer software code to perform, when the program element is executed on data processing means, the steps of:

storing a history buffer for containing an unencoded version of at least one previously encoded string;

comparing a string from said input data stream with said unencoded version of said at least one previously encoded string;

storing: an indicator that there exist at least two matches found by said first comparison

component, and tokens corresponding to said at least two matches;

summing potential token lengths to provide total potential token lengths;

comparing said total potential token lengths;

selecting a match corresponding to a shortest total token length to represent said string from said input data stream; and

emitting tokens representing said match corresponding to a shortest total token length.

13. A memory device storing computer program code to

perform, when the program element is executed on data processing means, the steps of:

storing a history buffer for containing an unencoded version of at least one previously encoded string;

comparing a string from said input data stream with said unencoded version of said at least one previously encoded string;

storing: an indicator that there exist at least two matches found by said first comparison component, and tokens corresponding to said at least two matches;

summing potential token lengths to provide total potential token lengths;

comparing said total potential token lengths;

selecting a match corresponding to a shortest total token length to represent said string from said input data stream; and

emitting tokens representing said match corresponding to a shortest total token length.